Bike 데이터 셋을 활용한 데이터 처리 및 시각화

In [2]:

```
import pandas as pd
```

In [3]:

```
train = pd.read_csv("../bike/train.csv", parse_dates=['datetime'])
test = pd.read_csv("../bike/test.csv", parse_dates=['datetime'])
```

In [4]:

```
train.columns
```

Out[4]:

```
Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
       'atemp', 'humidity', 'windspeed', 'casual', 'registered', 'count'],
      dtype='object')
```

In [5]:

```
test.columns
```

Out[5]:

```
Index(['datetime', 'season', 'holiday', 'workingday', 'weather', 'temp',
       'atemp', 'humidity', 'windspeed'],
      dtype='object')
```

In [6]:

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype         
---  ------      --------------  -----         
 0   datetime    10886 non-null  datetime64[ns]
 1   season      10886 non-null  int64         
 2   holiday     10886 non-null  int64         
 3   workingday  10886 non-null  int64         
 4   weather     10886 non-null  int64         
 5   temp        10886 non-null  float64       
 6   atemp       10886 non-null  float64       
 7   humidity    10886 non-null  int64         
 8   windspeed   10886 non-null  float64       
 9   casual      10886 non-null  int64         
 10  registered  10886 non-null  int64         
 11  count       10886 non-null  int64         
dtypes: datetime64[ns](1), float64(3), int64(8)
memory usage: 1020.7 KB
```

In [7]:

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6493 entries, 0 to 6492
Data columns (total 9 columns):
 #   Column      Non-Null Count  Dtype         
---  ------      --------------  -----         
 0   datetime    6493 non-null   datetime64[ns]
 1   season      6493 non-null   int64         
 2   holiday     6493 non-null   int64         
 3   workingday  6493 non-null   int64         
 4   weather     6493 non-null   int64         
 5   temp        6493 non-null   float64       
 6   atemp       6493 non-null   float64       
 7   humidity    6493 non-null   int64         
 8   windspeed   6493 non-null   float64       
dtypes: datetime64[ns](1), float64(3), int64(5)
memory usage: 456.7 KB
```

(실습1) 데이터를 알아가기 위한 여러가지 질문을 작성해 보자.

01. datetime은 언제부터 언제까지의 데이터일까?

02-A count와 temp는 어떤 상관관계가 있을까?

- 산점도(scatter plot)로 확인해 보기 - matplotlib 활용해 보기
- type은 점으로 표시
- 투명도를 0.2로 표현

02-B corr()를 활용하여 count와 다른 feature(특징)간의 상관계수를 확인해 보자.

- 가장 높은 상관관계를 갖는 순서로 정렬시켜보자.(pandas)
- 이를 수평 막대 그래프로 표시해 보자 - matplotlib 활용해 보기

03. 계절별 데이터를 확인 및 시각화 해 보자.

- season의 값의 종류와 count를 확인해 보기
- barplot 표시할 때, x축을 1,2,3,4만 표시되도록 하자.
- matplotlib 활용해 보기(

04. 쉬는날과 아닌날의 데이터는 얼마나 될까? 이를 시각화하기

- holiday의 값의 종류와 count를 확인해 보기
- 시각화 해보기(matplotlib 활용)

05. weather는 어떤 값을 갖고, 각각의 데이터의 수는 얼마나 될까?(시각화하기)

- weather의 값의 종류와 count를 확인해 보기
- 시각화 해보기(matplotlib 활용)

06. 아래의 값의 분포를 2행, 2열로 표시해 보자.

- temp의 값의 분포는 어떠할까?
- atemp의 값의 분포는 어떠할까?
- humidity의 값의 분포는 어떠할까?
- windspeed의 값의 분포는 어떠할까?
- 전체 그래프에 대한 제목을 달아보자(suptitle, 크기(size)=20))
- 각각의 그래프에 대한 x축 레이블을 넣어보자(크기는 17)
- 시각화 해보기(matplotlib 활용)

07. weather별 데이터의 비율은 어느정도 될까?

- 시각화 해보기(matplotlib 활용)
- 이에 대해서 pie 그래프로 나타내 보자.
- label은 한글로 '봄', '여름', '가을', '겨울'로 표시해 보자.

In [ ]: